

python for Computational Problem Solving

- pCPS - Set_Text_Files

Lecture Slides - Class #21_#22

Nitin V Pujari
Faculty, Computer Science
Dean - IQAC, PES University

pCPS Assignment Batches

```
,BatchId,ProjectBatch
0,pCPS_Assignment_Batch_ID_1,"('PES1202100893', 'PES1202100956', 'PES1202101345')"
```

1	pCPS_Assignment_Batch_ID_2	("('PES1202100862', 'PES1202101351', 'PES1202100999')")
2	pCPS_Assignment_Batch_ID_3	("('PES1202100802', 'PES1202100895', 'PES1202101314')")
3	pCPS_Assignment_Batch_ID_4	("('PES1202101342', 'PES2202100686', 'PES2202100705 ')")
4	pCPS_Assignment_Batch_ID_5	("('PES1202100868', 'PES1202100891', 'PES1202101354')")
5	pCPS_Assignment_Batch_ID_6	("('PES1202100884', 'PES1202100886', 'PES1202101033')")
6	pCPS_Assignment_Batch_ID_7	("('PES1202101027', 'PES1202101339', 'PES1202101054')")
7	pCPS_Assignment_Batch_ID_8	("('PES1202100959', 'PES1202100991', 'PES1202101048')")
8	pCPS_Assignment_Batch_ID_9	("('PES1202101466', 'PES1202101481', 'PES1202100838')")
9	pCPS_Assignment_Batch_ID_10	("('PES1202101050', 'PES1202101415', 'PES1202100970')")
10	pCPS_Assignment_Batch_ID_11	("('PES1202100960', 'PES1202100860', 'PES1202100967')")
11	pCPS_Assignment_Batch_ID_12	("('PES1202100974', 'PES1202100877', 'PES1202101330')")
12	pCPS_Assignment_Batch_ID_13	("('PES1202100801', 'PES1202101349', 'PES1202101480')")
13	pCPS_Assignment_Batch_ID_14	("('PES1202100803', 'PES1202101020', 'PES1202101513')")
14	pCPS_Assignment_Batch_ID_15	("('PES1202101315', 'PES1202101458', 'PES1202101460')")
15	pCPS_Assignment_Batch_ID_16	("('PES2202100680', 'PES1202100836', 'PES1202101014')")
16	pCPS_Assignment_Batch_ID_17	("('PES2202100695', 'PES1202101416', 'PES1202100930')")
17	pCPS_Assignment_Batch_ID_18	("('PES1202100816', 'PES1202101407', 'PES1202100890')")
18	pCPS_Assignment_Batch_ID_19	("('PES1202100829', 'PES1202101353', 'PES1202100841')")
19	pCPS_Assignment_Batch_ID_20	("('PES1202100789', 'PES1202101306', 'PES1202100830')")
20	pCPS_Assignment_Batch_ID_21	("('PES1202101329', 'PES1202100807', 'PES1202101038')")
21	pCPS_Assignment_Batch_ID_22	("('PES1202101041', 'PES1202100835', 'PES1202101051 ')")
22	pCPS_Assignment_Batch_ID_23	("('PES2202100627', 'PES1202100864', 'PES1202101358')")
23	pCPS_Assignment_Batch_ID_24	("('PES1202100928', 'PES1202101522', 'PES1202100953')")
24	pCPS_Assignment_Batch_ID_25	("('PES1202101538', 'PES1202101325')")

python for Computational Problem Solving Syllabus

Unit II: Collections & Basics of Functions - 12 Hours

Lists, Tuples , Dictionaries, Sets, Strings and text file manipulation: reading and writing files. Functions : Definition, call.

T1: 4.1 – 4.4 - Class #15, #16, #17, #18

T1: 9.1 – 9.2 - Class #19, #20, #21

T1: 5.1-5.2 - Class #25, #26

T1: 8.1, 8.2, 8.3 - Class #22, #23, #24

▼ 4 Lists

MOTIVATION

FUNDAMENTAL CONCEPTS

- ▶ 4.1 List Structures
- ▶ 4.2 Lists (Sequences) in Python
- ▶ 4.3 Iterating Over Lists (Sequences) in Python
- ▼ 4.4 More on Python Lists
 - 4.4.1 Assigning and Copying Lists
 - 4.4.2 List Comprehensions

▼ 9 Dictionaries and Sets

MOTIVATION

FUNDAMENTAL CONCEPTS

- ▶ 9.1 Dictionary Type in Python
- ▶ 9.2 Set Data Type

▼ 5 Functions

MOTIVATION

FUNDAMENTAL CONCEPTS

- ▶ 5.1 Program Routines
- ▶ 5.2 More on Functions

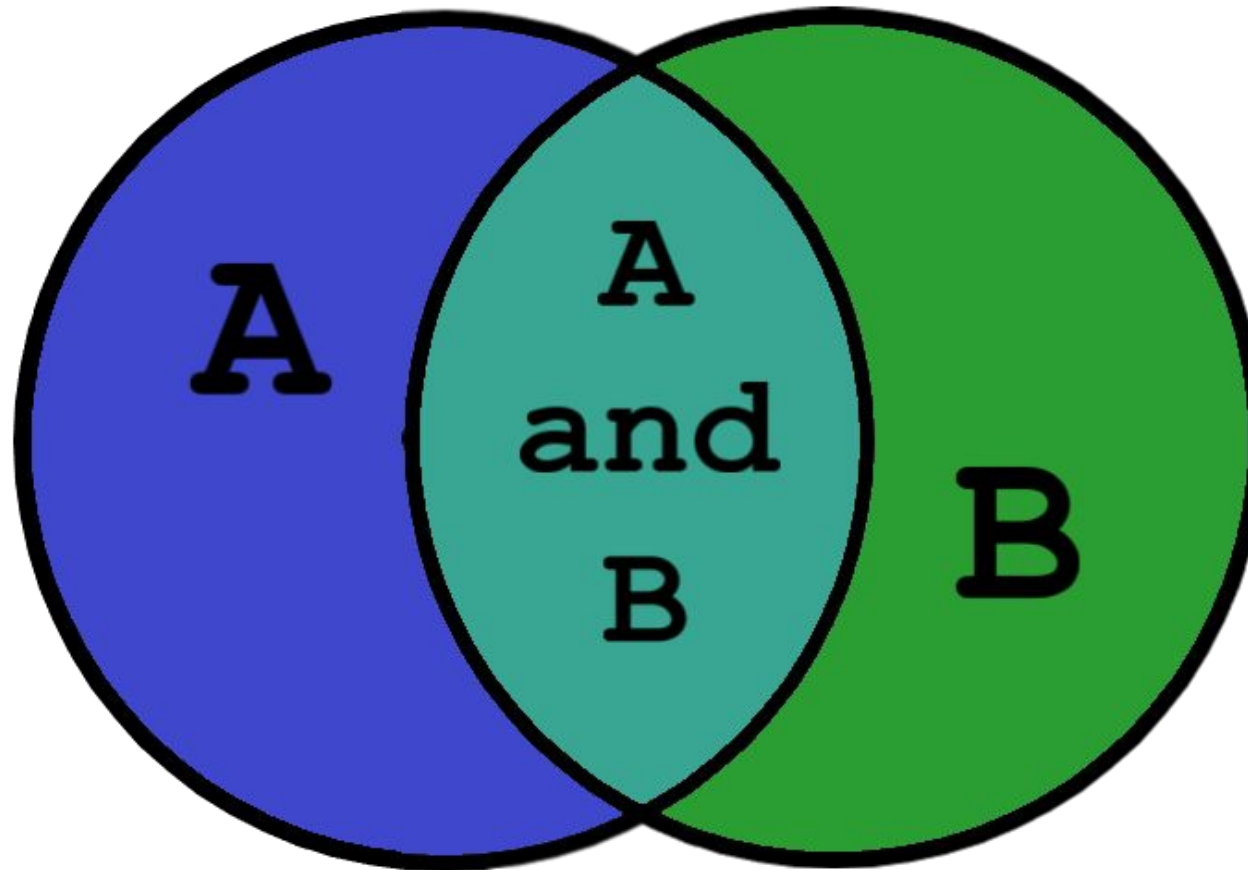
▼ 8 Text Files

MOTIVATION

FUNDAMENTAL CONCEPTS

- 8.1 What Is a Text File?
- ▶ 8.2 Using Text Files

▶ 8.3 String Processing



pCPS 9.2.1 set Data Type in python

- A set is a mutable data type with non duplicate, unordered values, providing the usual mathematical set operations as shown in the figure on the side

Set operator	Set A = {1,2,3} Set B = {3,4,5,6}		
membership	1 in A	True	<i>True if 1 is a member of set</i>
add	A.add(4)	{1,2,3,4}	<i>Adds new member to set</i>
remove	A.remove(2)	{1,3}	<i>Removes member from set</i>
union	A B	{1,2,3,4,5,6}	<i>Set of elements in either set A or set B</i>
intersection	A & B	{3}	<i>Set of elements in both set A and set B</i>
difference	A - B	{1,2}	<i>Set of elements in set A, but not set B</i>
symmetric difference	A ^ B	{1,2,4,5,6}	<i>Set of elements in set A or set B, but not both</i>
size	len(A)	3	<i>Number of elements in set (general sequence operation)</i>

pCPS 9.2.1 set Data Type in python

- One of the most commonly used set operators is the in operator for determining membership
- Note that the items in the set are not displayed in the order that they were defined.
- Sets, like dictionaries, do not maintain a logical ordering.
- The order that items are stored is determined by python, and not by the order in which they were provided.
- It is invalid and makes no sense to access an element of a set by index value

Set operator	Set A = {1,2,3} Set B = {3,4,5,6}		
membership	1 in A	True	True if 1 is a member of set
add	A.add(4)	{1,2,3,4}	Adds new member to set
remove	A.remove(2)	{1,3}	Removes member from set
union	A B	{1,2,3,4,5,6}	Set of elements in either set A or set B
intersection	A & B	{3}	Set of elements in both set A and set B
difference	A - B	{1,2}	Set of elements in set A, but not set B
symmetric difference	A ^ B	{1,2,4,5,6}	Set of elements in set A or set B, but not both
size	len(A)	3	Number of elements in set (general sequence operation)

pCPS 9.2.1 set Data Type in python

- A constructor is a special type of method (function) which is used to initialize the instance members of the class
- Constructors can be of two types.
 - Default constructor
 - Parameterized Constructor
 - Non-parameterized Constructor

pCPS 9.2.1 set Data Type in python

- The add and remove methods allow sets to be dynamically altered during program execution
- To define an initially empty set, or to initialize a set to the values of a particular sequence, the set constructor is used
- Note that set(), and empty braces are not used to create an empty set, since that notation is used to create an empty dictionary.
- Because sets do not have duplicate elements, adding an already existing item to a set results in no change to the set.
- Finally, there are two set types in python—the mutable set type, and the immutable frozenset type.
- Methods add and remove are not allowed on sets of frozenset type

pCPS 8 Text Files in python

- It has been estimated that 90% of all the data in the world has been generated in the last two years
- With the advent of the Internet, computers are not directly connected, but indirectly through routers that temporarily store and forward data toward its destination.
- The World Wide Web has made access to information easy and intuitive by the incorporation of hypertext— text that can be clicked on to retrieve more text—effectively making text “three-dimensional.”

Storage Technology	Example	Characteristics
Magnetic Storage	Hard drive	Nonvolatile
	Magnetic Tape Storage	Nonvolatile
Semiconductor Memory	Main memory	Volatile
	USB (Thumb) Drive	Nonvolatile
Optical Storage	CD, DVD	Nonvolatile

pCPS 8.1 Text File in python

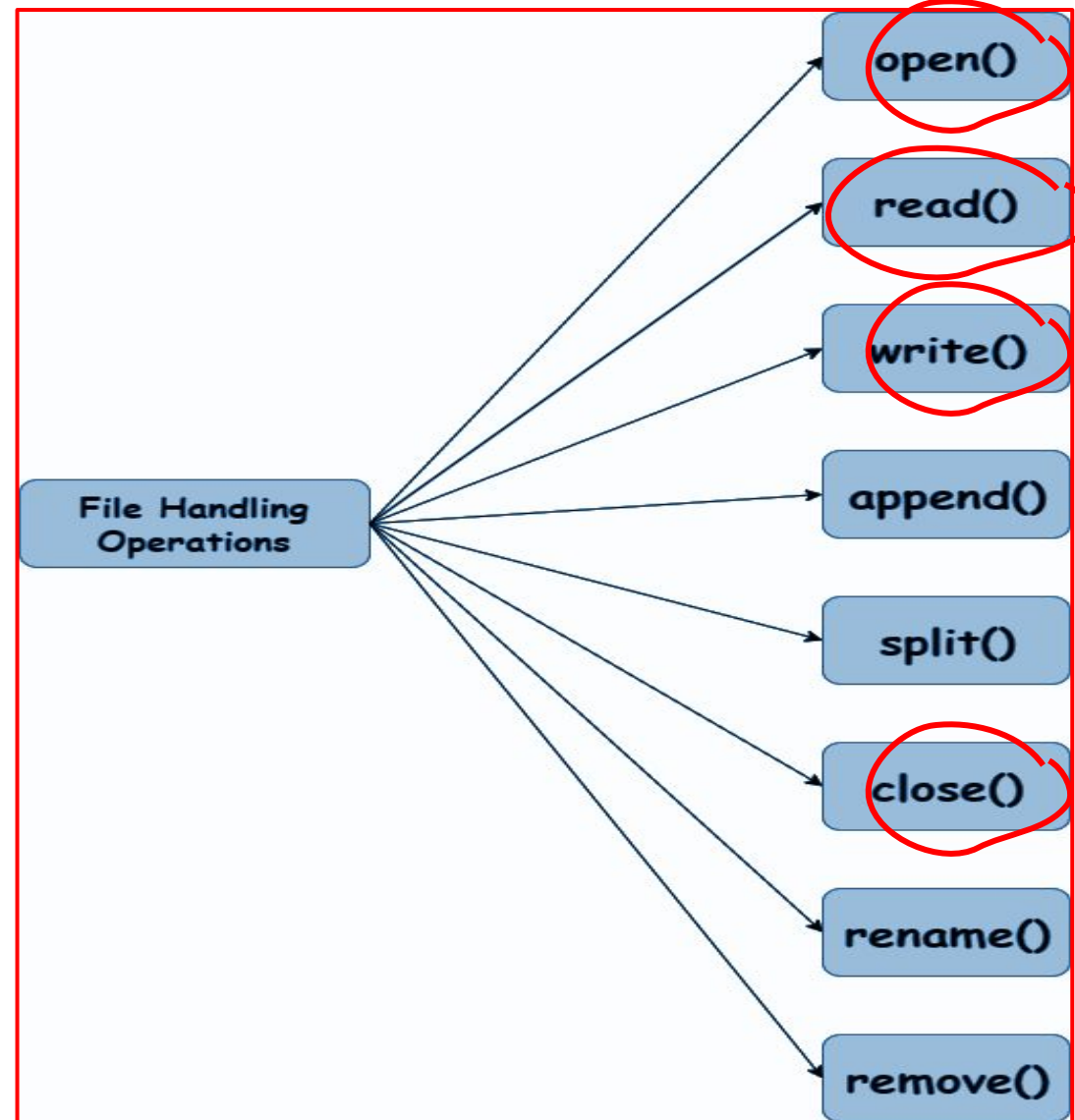
- A text file is a file containing characters, structured as individual lines of text.
- In addition to printable characters, text files also contain the nonprinting newline character, \n, to denote the end of each text line
- In contrast, binary files can contain various types of data, such as numerical values, and are therefore not structured as lines of text.
- Binary files can only be read and written via a computer program
- Any attempt to directly view a binary file will result in “garbled” characters on the screen.
- The purpose is not to cover all of types of files in python.
- Rather, we cover enough to be able to perform simple reading and writing of text files

pCPS 8.1 Text File in python

- A text file is a file containing characters, structured as individual lines of text.
- In addition to printable characters, text files also contain the nonprinting newline character, \n, to denote the end of each text line
- In contrast, binary files can contain various types of data, such as numerical values, and are therefore not structured as lines of text.
- Binary files can only be read and written via a computer program
- Any attempt to directly view a binary file will result in “garbled” characters on the screen.
- The purpose is not to cover all of types of files in python.
- Rather, we cover enough to be able to perform simple reading and writing of text files

pCPS 8.2 Using Text Files in python

- **Fundamental operations** of all types of **files** include **opening** a file, **reading** from a file, **writing** to a file, and **closing** a file.



pCPS 8.2.1 Opening Text Files in python

- All files must first be opened before they can be read from or written to.
- In python, when a file is successfully opened, a file object is created that provides methods for accessing the file.
- To open a file for reading, the built-in open function is used using the flag 'r'
- The first argument is the file name to be opened, 'Sample.txt'. The second argument, 'r', indicates that the file is to be opened for reading, which is optional.
- If the file is successfully opened, a file object is created and assigned to the provided identifier
- When opening a file for reading, there are a few reasons why an I/O error may occur

pCPS 8.2.1 Opening Text Files in python

- To open a file for writing, the built-in open function is used using the flag 'w'
- Note that, in this case, 'w' is used to indicate that the file is to be opened for writing. If the file already exists, it will be overwritten , starting with the first line of the file.
- When using a second argument of 'a', the output will be appended to an existing file instead.
- It is important to close a file that is written to, otherwise the tail end of the file may not be written to the file
- When opening files for writing, there is not much chance of an I/O error occurring.
- The provided file name if it does not need to exist and is being created or overwritten.
- The only error that may occur is if the file system such as the hard disk is full or you may not have have required permissions

pCPS 8.2.2 Reading Text Files in python

- The readline method returns as a string the next line of a text file, including the end-of-line character, \n.
- When the end-of-file is reached, it returns an empty string as demonstrated in the while loop in the figure on the side
- It is also possible to read the lines of a file by use of the for statement
- Using a for statement, all lines of the file will be read one by one.
- Using a while loop, however, lines can be read until a given value is found

Text File myfile.txt

```
Line One\n
Line Two\n
Line Three\n
```

```
input_file = \
    open('myfile.txt','r')
empty_str = ''

line = input_file.readline()

while line != empty_str:
    print(line)
    line = input_file.readline()

input_file.close()
```

Screen Output

```
Line One
Line Two
Line Three
```

```
input_file = open('myfile.txt','r')
for line in input_file:
    print(line)
```

pCPS 8.2.3 Writing to a Text File in python

- The write method is used to write strings to a file
- This code copies the contents of the input file, 'myfile.txt', line by line to the output file, 'myfile_copy.txt'.
- In contrast to print when writing to the screen, the write method does not add a newline character to the output string
- Thus, a newline character will be output only if it is part of the string being written. In this case, each line read contains a newline character

Text File
myfile.txt

```
Line One\nLine Two\nLine Three\n
```

```
empty_str = ''  
input_file = open('myfile.txt','r')  
output_file = open('myfile_copy.txt','w')  
  
line = input_file.readline()  
  
while line != empty_str:  
    output_file.write(line)  
    line = input_file.readline()  
  
output_file.close()
```

Text File
myfile_copy.txt

```
line one  
line two  
line three
```


pCPS 8.2.3 Writing to a Text File in python

- When writing to a file, data is first placed in an area of memory called a buffer .
- Only when the buffer becomes full is the data actually written to the file.
- This makes reading and writing files more efficient.
- Since the last lines written may not completely fill the buffer, the last buffer's worth of data may not be written. The close() method flushes the buffer to force the buffer to be written to the file

Text File
myfile.txt

```
Line One\n
Line Two\n
Line Three\n
```

```
empty_str = ''
input_file = open('myfile.txt','r')
output_file = open('myfile_copy.txt','w')

line = input_file.readline()

while line != empty_str:
    output_file.write(line)
    line = input_file.readline()

output_file.close()
```

Text File
myfile_copy.txt

```
line one
line two
line three
```

pCPS 8.2.3 Writing to a Text File in python

Self-Test Questions

1. Only files that are written to need to be opened first. (TRUE/FALSE)
2. Indicate which of the following reasons an IOError (exception) may occur when opening a file.
(a) Misspelled file name (c) File not found in directory searched
(b) Unmatched uppercase and lowercase letters
3. Which one of the following is true?
(a) When calling the built-in open function, a second argument of 'r' or 'w' must always be given
(b) When calling the built-in open function, a second argument of 'r' must always be given when opening a file for reading
(c) When calling the built-in open function, a second argument of 'w' must always be given when opening a file for writing
4. Which one of the following is true?
(a) There is more chance of an I/O error when opening a file for reading.
(b) There is more chance of an I/O error when opening a file for writing.
5. The readline method reads every character from a text file up to and including the next newline character '\n'. (TRUE/FALSE)
6. It is especially important to close a file that is open for writing. (TRUE/FALSE)



THANK YOU



Nitin V Pujari
Faculty, Computer Science
Dean - IQAC, PES University
nitin.pujari@pes.edu

For Course Digital Deliverables visit www.pesuacademy.com